

#2f  
6/17/02  
DH

Attorney Docket No. 1454.1225

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Rudolf CASPARI

Application No.:

Group Art Unit: To be assigned

Filed: March 7, 2002

Examiner: To be assigned

For: PROVIDING HELP INFORMATION IN A SPEECH DIALOG SYSTEM

1c996 U.S. PRO  
10/091584  
03/07/02

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)  
herewith a certified copy of the following foreign application:

German Patent Application No. 101 10 977.6

Filed: March 7, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing  
date(s) as evidenced by the certified papers attached hereto, in accordance with the  
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 7, 2002

By: Mark J. Henry  
Mark J. Henry  
Registration No. 36,162

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001  
(202) 434-1500



1c996 U.S. PTO  
10/091584  
03/07/02

## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 101 10 977.6

**Anmeldetag:** 7. März 2001

**Anmelder/Inhaber:** Siemens AG, München/DE

**Bezeichnung:** Bereitstellen von Hilfe-Informationen in einem Sprachdialogsystem

**IPC:** G 10 L 15/22

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 3. Dezember 2001  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

*Jerofsky*

## Beschreibung

## Bereitstellen von Hilfe-Informationen in einem Sprachdialogsystem

5

Die Erfindung betrifft ein Verfahren, ein Computerprogrammprodukt, einen Datenträger und eine Einrichtung zum Bereitstellen von Hilfe-Informationen für einen Benutzer in einem Sprachdialogsystem zum Bedienen einer Hintergrundapplikation.

10

Anwendungen bzw. Hintergrundapplikationen, wie zum Beispiel ein technisches Gerät der Unterhaltungselektronik, ein telefonisches Auskunftssystem (Bahn, Flug, Kino, etc.), ein computergestütztes Transaktionssystem (Homebanking-System, elektronische Warenbestellung, etc.) werden in zunehmendem Maße über ein Sprachdialogsystem als Zugangssystem (User Interface) bedient. Derartige Sprachdialogsysteme können in Hardware, Software oder einer Kombination daraus realisiert werden.

20

Damit ein Benutzer seine Anfragen in gesprochener natürlicher Sprache stellen kann, ist die Verwendung von Spracherkennern erforderlich. Spracherkennungsverfahren sind beispielsweise aus US 6 029 135, US 5 732 388, DE 196 36 739 C1, DE 197 19 381 C1 und DE 199 56 747 C1 bekannt. Diese verfügen jedoch häufig über einen stark restringierten Wortschatz.

25

Bei der Bedienung mittels Sprache treten daher folgende Probleme auf:

30

- a) der Benutzer weiß nicht, was er sprechen darf und
- b) der Benutzer kennt nicht das konzeptuelle Modell, auf dem die Hintergrundapplikation beruht.

35

Im Fall a) kommt es leicht zu Fehlerkennungen des Spracherkenners. Falls der Zugang zum System über geschriebene Sprache erfolgt, tritt dieser Nachteil nur dann

auf, wenn das System versucht, alle Wörter der Eingabe zu interpretieren. Allerdings bietet der Zugang über gesprochene Sprache eine wesentlich attraktivere Benutzerschnittstelle, so dass das Problem in der Praxis eine große Rolle spielt.

5

Im Fall b) kann der Benutzer das System ohne vorheriges Studieren einer Bedienungsanleitung bestenfalls eingeschränkt benutzen. Eine Bedienungsanleitung muss aber nicht verfügbar sein, z.B. wenn sich der Benutzer über einen telefonischen Sprachdialog eine Information verschaffen möchte. Darüber hinaus ist das Lesen einer Bedienungsanleitung in jedem Fall mit erhöhtem Aufwand für den Benutzer verbunden. Dies vermindert insbesondere die Akzeptanz bei der Bedienung komplexer Systeme. Problem b) tritt auch auf, wenn der Zugang über geschriebene Sprache erfolgt.

15

Daher wurden verschiedene Hilfesysteme für Sprachdialogsysteme entwickelt.

- 20 IVR-(Interactive Voice Response)-Systeme bieten kontextsensitive Hilfe. Sie sind als Sprachmenüs nach folgendem Muster konstruiert:
- "Wenn sie A wollen, sagen Sie  $f(A)$ .", wobei  $f(.)$  eine Funktion darstellt, z.B. eine, die die Zahlen einer
  - 25 Telefontastatur als Werte ausgibt.
  - "Wenn Sie B wollen, sagen Sie  $f(B)$ ."
  - "Wenn Sie Y wollen, sagen Sie  $f(Y)$ ."

25

In jeder Situation erhält der Benutzer einen Überblick über seine Möglichkeiten. Bei Auswahl einer der Möglichkeiten wechselt das System zur nächst tieferen Ebene und gibt dort wieder einen Überblick über die zur Verfügung stehenden Möglichkeiten.

30

35 Der Nachteil von IVR-Systemen besteht darin, dass sie zwar für einen ungeübten Benutzer akzeptabel sind, in jedem Fall aber zu langwierigen und umständlichen Dialogen führen, da

die Dialog-Initiative in jedem Fall vom System ausgeht, was für geübte Benutzer nicht hinnehmbar ist. Die Benutzerführung ist starr.

- 5 Infolge der Unzulänglichkeiten der IVR-Systeme wurde die Dialogmaschine Diane entwickelt (DE 196 15 693 C1). Sie erlaubt einen Dialog mit gemischter Initiative. Diane geht davon aus, dass zunächst der Benutzer die Initiative besitzt. Ein geübter Benutzer kann einen gezielten Befehl erteilen  
10 oder eine gezielte Anfrage oder Hilfeanfrage stellen, ohne dass ihm seine Möglichkeiten vorher langwierig aufgezählt werden müssten. Auf diese Weise wird ein direkter Zugriff auf alle Möglichkeiten des Systems gewährleistet, insbesondere auch auf solche Hilfeangebote, die bei IVR-Systemen in  
15 tieferen Ebenen angesiedelt sind.

- Ein Dialog wird nur erforderlich, wenn die Anfangsäußerung des Benutzers (i) unvollständig oder (ii) mehrdeutig ist oder (iii) im Widerspruch zu den Möglichkeiten der  
20 Hintergrundapplikation steht. Falls einer der genannten drei Fälle auftritt, ergreift Diane die Initiative und führt mit dem Benutzer einen Klärungsdialog, um die gewünschte Intention des Benutzers zu ermitteln und fehlende Wissensseinheiten zu erfragen.

- 25 Diane benutzt ein abstraktes Aufgabenmodell, welches auf folgenden Prinzipien beruht, die auch für die Einbeziehung eines Hilfesystems vorausgesetzt werden:

- P1) Die Hintergrundapplikation kann aufgefasst werden als  
30 eine endliche Menge von Transaktionen T1, T2, ..., Tn.

- P2) Jede Transaktion hat einen Namen und besitzt eine endliche Menge (die auch leer sein kann) von Informationsparametern I1, I2, ..., Im. Diese Parameter müssen dem System bekannt sein, damit die Transaktion  
35 ausgeführt werden kann.

- P3) Zu jedem Parameter gehört eine Grammatik, die dazu dient, den Parameter im Dialog zu erfassen.

Der Benutzer kann die gewünschte Transaktion und die dazugehörigen Parameter in einem Satz benennen oder nicht. Im ersten Fall kann die Transaktion sofort durchgeführt werden.

- 5 Im zweiten Fall werden die noch unbekannten Parameter im Dialog erfasst.

- 10 Falls durch die Anfangsäußerung des Benutzers keine Transaktion eindeutig bestimmbar ist, führt das System automatisch einen Klärungsdialog zur Ermittlung der gewünschten Aktion durch. Gleiches gilt für unklare oder unvollständige Parametereingaben.

- 15 Zur Verdeutlichung betrachte man folgendes Beispiel: Die Hintergrundapplikation realisiert das folgende Aufgabenmodell, bestehend aus Transaktionen und den in Klammern aufgeführten Parametern:

- Zugauskunft (Startort, Zielort, Startzeit, Abfahrtsdatum)
- Flugauskunft (Startort, Zielort, Startzeit, Abfahrtsdatum)
- 20 - Geldüberweisung (Geldbetrag, Empfängerkonto, Empfänger Bank bzw. Bankleitzahl)
- Geldabhebung (Geldbetrag, Kontonummer, Bankleitzahl, PIN)
- Aktienkauf (Anzahl, Gesellschaft, Limit)
- Aktienverkauf (Anzahl, Gesellschaft, Limit)
- 25 - Senderaufnahme (Sender, Startzeit, Stopzeit)
- Sendungsaufnahme (Sendung, Datum)
- Anruf (Telefonnummer)

- 30 Die genannten drei Prinzipien führen bei dem Dialogsystem Diane zu folgenden Zuständen, in denen das System vom Benutzer eine Eingabe erwartet:

- Zustand a): Es ist noch keine Transaktion ausgewählt und die Transaktionen T1, T2, ..., Ti sind noch möglich.
- Zustand b): Das System hat eine Transaktion ausgewählt und 35 erfragt vom Benutzer einen Parameterwert.
- Zustand c): Das System hat eine ja-nein-Frage gestellt.

Die Dialogmaschine Diane bietet jedoch keine kontextsensitive Hilfe.

5 In jüngster Zeit wurde die Sprache VoiceXML definiert, mit dem Ziel, das Telefon als natürlichsprachliche Zugangsmöglichkeit für Hintergrundapplikationen im Internet nutzen zu können.

10 VoiceXML erlaubt die natürlichsprachliche Navigation in Dokumenten, die über das Internet von einem Dokumentenserver zu Verfügung gestellt werden. Von einem Wurzeldokument ausgehend, kann der Benutzer in diesem Dokument einen Dialog führen oder per Sprachkommando zu anderen Dokumenten springen. In jedem auf diese Weise erreichten Dokument können  
15 dann Dialoge ablaufen, die auf Grammatiken beruhen, welche in diesem Dokument definiert sind.

Die Sprache VoiceXML bietet ferner ein Sprachkonstrukt für Hilfe. VoiceXML stellt Hilfetags zur Verfügung, bei denen der  
20 Programmierer an jeder Stelle im Dialog auf Hilfeanfragen des Benutzers reagieren kann. Diese Hilfe ist jedoch nur als Möglichkeit vorgesehen und nicht allgemein in das Bedienmodell integriert. Sie wird vom Programmierer direkt an der gewünschten Stelle in den Code integriert. Das führt  
25 dazu, dass das System die Hilfeanfragen des Benutzers nicht immer beantworten kann. Es besteht außerdem keine Verknüpfung zwischen den Hilfe-Prompts und den im Dialog verwendeten Grammatiken.

30 Nachteilig an der Sprache VoiceXML ist ferner, dass sie von Hand programmiert werden muss.

Daneben wurden sogenannte SpeechObjects entwickelt. Das sind wiederverwertbare, abgekapselte Dialogteile. Sie bestehen aus  
35 einer Grammatik zum Parsen des Benutzerinputs, Prompts für die Ausgabe und einer Ablauflogik. Sie werden dazu benutzt, um komplexere Dialoge aufzubauen. Mittels solcher

SpeechObjects kann auch eine Hilfe von Hand programmiert werden. Dies erfordert jedoch einen hohen Programmieraufwand.

5 Aufgabe der Erfindung ist es, ein Hilfesystem zu schaffen, das den Benutzer bei der Bedienung eines Sprachdialogsystems zum Bedienen einer Hintergrundapplikation unterstützt.

Diese Aufgabe wird durch die unabhängigen Ansprüche gelöst. Vorteilhafte Weiterbildungen der Erfindung sind in den  
10 Unteransprüchen gekennzeichnet.

Ferner wird die Aufgabe durch ein Computerprogrammprodukt mit maschinenlesbarem Programmcode zur Durchführung des erfindungsgemäßen Verfahrens gelöst. Unter einem  
15 Computerprogrammprodukt wird dabei das Computerprogramm als handelbares Produkt verstanden, in welcher Form auch immer, z.B. maschinenlesbar auf Papier, auf einem computerlesbaren Datenträger, über ein Netz verteilt, etc.

20 Außerdem wird die Aufgabe durch einen Datenträger gelöst, auf dem ein Computerprogramm gespeichert ist, das nach einem Laden in den Hauptspeicher das erfindungsgemäße Verfahren ausführt.

25 Im Folgenden wird die Erfindung ausführlich erläutert.

Für eine Hintergrundapplikation, welche nach den oben genannten Prinzipien P1, P2, P3 modelliert ist und ein Dialogsystem, welches die oben genannten Zustände a), b), c)  
30 kennt, wird in einem ersten Schritt eine flache kontextsensitive Hilfe erzeugt.

Zunächst wird eine Hilfegrammatik definiert. Ein einfachstes Beispiel für eine Hilfegrammatik besteht beispielsweise  
35 darin, dass das Sprachdialogsystem z.B. die Äußerungen "Hilfe!", "Bitte helfen Sie mir." oder "Was kann ich tun?" versteht und mit einem Prompt, d.h. mit einer Aussage,



antwortet. Allgemein gesprochen ist ein Prompt eine Antwort bzw. eine Äußerung des Systems. Die Prompts des Systems werden durch eine Hilfeanfrage auf Initiative des Benutzers ausgelöst.

5

Im Folgenden werden eine Reihe von Prompts definiert, die die Auswahl von Transaktionen oder die Eingabe von Parameterwerten unterstützen. Weiter unten werden Beispiele für die Verwendung der Prompts gegeben.

10

Für jede Transaktion definiert man einen Transaktionsprompt. Beispiele für Transaktionsprompts, d.h. von Hilfe-Aussagen (rechte Spalte) des Systems zu den einzelnen Transaktionen (linke Spalte), sind:

15

Zugauskunft	"eine Zugauskunft einholen"
Flugauskunft	"eine Flugauskunft einholen"
Geldüberweisung	"Geld überweisen"
Geldabhebung	"Geld abheben"
20 Aktienkauf	"Aktien kaufen"
Aktienverkauf	"Aktien verkaufen"
Senderaufnahme	"einen Sender aufnehmen"
Sendungsaufnahme	"eine Sendung aufnehmen"
Anruf	"jemanden anrufen"

25

Die Transaktionsprompts bestehen also z.B. aus Objekt und Infinitiv.

Außerdem definiert man einen pauschalen Hilfeprompt.

30

Beispiele für pauschale Hilfeprompts sind: "Sie können: ..." oder "Sagen Sie eine der folgenden Möglichkeiten: ...?", gefolgt jeweils von der Aufzählung aller Möglichkeiten, ausgedrückt durch die jeweiligen Transaktionsprompts. Die pauschalen Hilfeprompts bestehen also z.B. aus Subjekt und

35 Modalverb.

Aus dem pauschalen Hilfeprompt und den Transaktionsprompts wird durch Hintereinanderschaltung ein vollständiger Satz mit Subjekt, Prädikat und Objekt erzeugt: "Sie können jemanden anrufen." Dabei wird das Prädikat aus der Kombination von  
5 Modalverb und Infinitiv gebildet.

Es wäre auch denkbar, auf den pauschalen Hilfeprompt zu verzichten und die Transaktionsprompts gleich in der Form "Sie können jemanden anrufen." zu definieren.

10

Für jeden Parameter einer Transaktion definiert man einerseits einen Parameterprompt, mit dem das System fehlende Werte für Parameter erfragt, z.B. "Wie lautet Ihr Abfahrtsort?" oder "Nennen Sie den Abfahrtstort."

15

Zusätzlich wird zu jedem Parameter ein Hilfeprompt definiert, und zwar entweder ein Parameter-Hilfeprompt oder ein Möglichkeitsprompt. Welcher Prompt definiert und gewählt wird, wird weiter unter im Detail erläutert.

20

Mittels des Möglichkeitsprompts werden alle möglichen Werte für den jeweiligen Parameter aufgezählt. Der Parameter-Hilfeprompt, andererseits, gibt an, in welcher Form der Benutzer einen Wert für den Parameter eingeben kann, um im  
25 Beispiel zu bleiben z.B.: "Nennen Sie als Abfahrtsort einen Ort in Deutschland."

Beispiele für Parameterprompts (rechte Spalte, jeweils erste Zeile) und für Parameter-Hilfeprompts (rechte Spalte, jeweils  
30 zweite Zeile) zu den einzelnen Parametern (linke Spalte) sind:

Startort	"Wie lautet Ihr Abfahrtsort?" "Nennen Sie als Abfahrtsort einen Ort in Deutschland."
35 Startzeit	"Wann wollen Sie abfahren?" "Sagen Sie die Abfahrtszeit, z.B. 17 Uhr 45."

Geldbetrag "Welchen Betrag wollen Sie überweisen?"  
"Nennen Sie den zu überweisenden Betrag, z.B.  
400 Euro 60."

5 Kontonummer "Wie lautet Ihre Kontonummer?" oder "Nennen  
Sie Ihre Kontonummer."  
"Sagen Sie Ihre Kontonummer als Folge von  
Ziffern."

10 Sender "Welchen Fernsehsender wollen Sie aufnehmen?"  
"Nennen Sie den Fernsehsender, den Sie  
aufnehmen wollen."

Datum "An welchem Tag läuft die Sendung?"  
"Nennen Sie das Datum in folgendem Format:  
z.B. 12. Februar."

15 Anzahl Aktien "Wie viel Stück?"  
"Nennen Sie die Anzahl der Aktien, die Sie  
kaufen wollen in der Form einer natürlichen  
Zahl, z.B. 500."

20 Statt des Parameter-Hilfeprompts kann der Möglichkeitsprompt  
für die Parametereingabe definiert werden, mit dessen Hilfe  
die für den jeweiligen Parameter möglichen Parameterwerte  
aufgelistet werden. Der Möglichkeitsprompt lautet z.B.:  
"Sagen Sie eine der folgenden Möglichkeiten: ...". Gefolgt  
wird er von der Auflistung aller Möglichkeiten. Ein Beispiel  
25 für die Verwendung des Möglichkeitsprompts beim Aktienkauf,  
wenn als Parameter das Unternehmen, dessen Aktien gekauft  
werden sollen, eingegeben werden muss, ist: "Sagen Sie eine  
der folgenden Möglichkeiten: BASF, Siemens, Deutsche Bank,  
...". Die Möglichkeiten werden aus der Grammatik der  
30 jeweiligen Parameter generiert (siehe unten).

Ferner definiert man einen ja-nein-Hilfeprompt. Ein Beispiel  
für einen ja-nein-Hilfeprompt ist: "Bitte beantworten Sie die  
Frage "...?" mit 'Ja' oder 'Nein'." , wobei die zuletzt  
35 gestellte Frage wiederholt wird.

Für Klärungsdialoge wird zusätzlich ein Frageprompt definiert. Dieser lautet z.B.: "Wollen Sie ...?" Wird er durch einen Transaktionsprompt ergänzt, ergibt sich ein vollständiger Satz mit Subjekt, Prädikat und Objekt: "Wollen  
5 Sie ein Zugauskunft einholen?"

Im folgenden wird der Einsatz der definierten Prompts geschildert.

10 Auf eine Anfrage des Systems an den Benutzer (z.B. "Sie wünschen?" nach dem Einschalt- bzw. Startvorgang, oder "Geben Sie bitte die Parameter (der ausgewählten Transaktion) an.") kann der Benutzer entweder einen geeigneten Befehl sprechen oder eine kontextsensitive Hilfe anfordern. Dazu äußert er  
15 eine der Formen, die von der Hilfegrammatik verstanden wird. Wenn die Benutzeräußerung durch die Hilfegrammatik erfasst wird, reagiert das System - das sich genau in einem der Zustände a), b) oder c) befindet - wie folgt:

20 In Zustand a):

Ist noch keine Transaktion ausgewählt, so kann der Benutzer entweder eine Transaktion - gemäß der dafür vorgesehenen Grammatik - auswählen, oder Hilfe erfragen. Erfragt er Hilfe  
25 und sind die Transaktionen T1, T2, ..., Ti noch möglich, so reagiert das System mit der Ausgabe: "(Pauschaler Hilfprompt): (Transaktionsprompt von T1), (Transaktionsprompt von T2), ..., (Transaktionsprompt von Ti)." Der pauschale Hilfprompt "Sie können: ..." wird  
30 ergänzt durch die Auflistung der Transaktionsprompts. Im Zustand a) des Systems hört der Benutzer somit z.B.: "Sie können: eine Zugauskunft einholen, eine Flugauskunft einholen, Geld überweisen, Geld abheben, ...".

35 In Reaktion darauf äußert der Benutzer: "Eine Zugauskunft einholen" oder "Zugauskunft" oder jeden beliebigen anderen, grammatisch zulässigen Transaktionsaufruf. Ebenso gut kann er

den Hilfeaufruf wiederholen, wenn ihm seine Entscheidung oder seine Möglichkeiten noch nicht klar sind.

5 In einem Klärungsdialog (gemäß DE 196 15 693 C1, s.o.) des Zustands a) des Systems nach einer unklaren Eingabe hört der Benutzer z.B. die Frage: "(Frageprompt) + (Transaktionsprompt)", wobei der Transaktionsprompt der vom System als am wahrscheinlichsten ermittelten Transaktion ausgegeben wird. Ein Beispiel lautet somit: "Wollen Sie eine  
10 Zugauskunft einholen?" Die Antwort des Benutzers darauf lautet "ja" / "ja bitte" / "nein" oder er äußert den Befehl: "Eine Zugauskunft einholen" oder "Zugauskunft" oder jeden beliebigen anderen, grammatisch zulässigen Transaktionsaufruf. Ebenso gut kann er einen Hilfeaufruf  
15 durchführen, wenn ihm seine Entscheidung oder seine Möglichkeiten noch nicht klar sind. Es werden dann alle zur Verfügung stehenden Transaktionen nach dem oben angegebenen Schema aufgezählt. In einer Variante des Klärungsdialogs weist das System - falls sich der Benutzer nicht äußert -  
20 nach einer kurzen Wartezeit auf andere mögliche Transaktionen hin.

25 Stehen nur sehr wenige Transaktionen zur Verfügung, so kann das System statt der Aufzählung der Möglichkeiten auch jeweils einen Ja/Nein-Dialog in der Form "(Frageprompt) + (Transaktionsprompt von Ti)" für jede mögliche Transaktion führen: "Wollen Sie eine Zugauskunft einholen?" Nach diesen Fragen wartet das System auf eine Antwort. Nach einer Weile ohne Antwort kann das System die nächste Transaktion durch  
30 eine Frage vorschlagen.

Ein Hilfeaufruf wird in dieser Situation durch einen ja-nein-Hilfeprompt beantwortet.

35 Damit das System nicht zu viele Möglichkeiten in der Form "(Frageprompt) + (Transaktionsprompt von Ti)" hintereinander ausgibt und anschließend auf eine Antwort wartet und dadurch

den Benutzer ermüdet, wird die Anzahl der auf diese Weise hintereinander auszugebenden Möglichkeiten begrenzt. Dazu definiert man eine natürliche Zahl D, die als Dialogschwelle bezeichnet wird. Ein Vergleich der Anzahl der Möglichkeiten mit D entscheidet darüber, ob die zur Verfügung stehenden Möglichkeiten in der Form "(Frageprompt) + (Transaktionsprompt von Ti)" oder ob alle Möglichkeiten geschlossen mittels des pauschalen Hilfeprompts ("Sagen Sie eine der folgenden Möglichkeiten: ...") ausgegeben werden. Sinnvolle Werte für D sind etwa 2 oder 3. Liegen mehr als 2 oder 3 Möglichkeiten vor, so wird der pauschale Hilfeprompt gewählt.

In Zustand b):

Der Benutzer hat eine Transaktion des Systems ausgewählt, z.B. Aktien kaufen. Nun erwartet das System vom Benutzer die Eingabe mindestens eines Parameterwerts. Der Benutzer kann z.B. sagen: "Ich möchte 200 Siemens Aktien kaufen.", womit er dem System zwei Parameter übergeben hat: den Namen des Unternehmens, dessen Aktien gekauft werden sollen, und die Anzahl der zu kaufenden Aktien.

Sollte die Eingabe von Parametern unklar sein, führt das System einen Klärungsdialog mit dem Benutzer durch (siehe DE 196 15 693 C1).

Äußert sich der Benutzer nicht innerhalb einer gewissen Zeit, so ergreift das System die Initiative und erfragt die Parameterwerte vom Benutzer mittels der Parameterprompts.

Der Benutzer spricht dann entweder den Parameterwert oder die Parameterwerte in der für diese Transaktion bzw. Parameter definierten Grammatik, oder er erfragt eine Hilfe zur Eingabe des Parameterwerts. Für die Erzeugung des Hilfeprompts gilt es, zwei Fälle zu unterscheiden:

Fall a): Die Parametergrammatik hat die Generierungseigenschaft, d.h. mit der Grammatik ist eine Liste aller Möglichkeiten für die Parametereingabe verknüpft, z.B. die Liste aller Unternehmen des DAX. Die Grammatik umfasst im einfachsten Fall also etwa die Liste "BASF, Siemens, Deutsche Bank, ...". (Je nach verwendetem Grammatikformalismus können die Möglichkeiten auch automatisch aus der Grammatik berechnet werden.) Das System äußert dann den Möglichkeitsprompt und listet alle Möglichkeiten auf, die sich aus der Grammatik ergeben. Im Beispiel des Aktienkaufs ergibt sich z.B. folgender Dialogausschnitt: (Parameterprompt:) "Von welcher Gesellschaft möchten Sie Aktien kaufen?", "Hilfe!", (Möglichkeitsprompt:) "Sagen Sie eine der folgenden Möglichkeiten: BASF, Siemens, Deutsche Bank, ...".

Fall b): Die Parametergrammatik hat nicht die Generierungseigenschaft, d.h. es ist praktisch unmöglich, alle Möglichkeiten für den benötigten Parameterwert aufzulisten. Ein Beispiel dafür ist die Uhrzeit. Das System äußert dann den Parameter-Hilfeprompt, z.B.: "Sagen Sie die Abfahrtszeit, z.B. 17 Uhr 45." Der Benutzer kann nun den Parameterwert in der Grammatik gemäß den vom System gegebenen Beispielen durch seine Sprache eingeben. Oder er wiederholt die Hilfeanfrage.

In Zustand c):

Das System hat eine ja-nein-Frage gestellt und erwartet "Ja" oder "Nein" als Antwort des Benutzers. Hat der Benutzer die Orientierung verloren, kann er die Hilfe aufrufen. Das System äußert den ja-nein-Hilfeprompt unter Wiederholung der Frage. Anschließend kann der Benutzer antworten.

Von den IVR-Systemen ist das Prinzip bekannt, dem Benutzer an jeder Stelle des Dialogs Angaben über die zur Verfügung stehenden Möglichkeiten zu machen. Dieses Prinzip wird

erfindungsgemäß mit einer Sprache verknüpft, die zur Modellierung der zugrundliegenden Hintergrundapplikation dient, indem die Sprache um Hilfeslots erweitert wird. Das System generiert dann aus den Hilfeslots und dem

5 Kontextwissen die jeweils angemessene Hilfe.

Grundsätzlich kann man zwischen statischen und dynamischen Hilfesystemen unterscheiden. Statische Hilfesysteme (z.B. Microsoft Hilfe für Word) geben dem Benutzer Hilfe zu von ihm

10 formulierten Themen. Sie sind nur benutzbar, falls der Benutzer das konzeptuelle Modell des Hintergrundsystems einigermaßen kennt, da er eine gezielte Frage stellen muss. Statische Hilfe kann in jeder Situation angefordert werden und liefert situationsunabhängig immer das gleiche Ergebnis,

15 sofern die Anfrage sich nicht ändert.

Dynamische Hilfesysteme können unabhängig von statischen Hilfesystemen und gleichzeitig mit diesen existieren. Sie unterstützen den Benutzer während der Ablaufphase eines

20 komplexen Bedienprozesses (der hier über einen Sprachdialog realisiert wird) in der jeweiligen Situation kontextabhängig. Charakteristisch ist hier, dass der Benutzer keine spezielle Frage stellt, sondern sich über die generelle Frage "Was ist möglich?" einen Überblick über die gerade gültigen

25 Möglichkeiten verschaffen kann. Der Benutzer braucht kein konzeptuelles Modell der Aufgabe zu besitzen. Vielmehr erlernt er dieses, indem er über ein globales Hilfefkommando die im jeweiligen Kontext gültigen Optionen vom System vermittelt bekommt.

30

Dynamische Hilfesysteme sind nur einsetzbar, wenn das System selbst zu jedem Zeitpunkt Zugriff auf das komplette Wissen hat, das zur Bedienung erforderlich ist und dieses Wissen auch hinreichend strukturiert ist.

35



Der vom System generierte Hilfemechanismus ist uniform und damit für den Endbenutzer leicht verstehbar. Ein globales Hilfefkommando ist leicht erlernbar.

- 5 Bei dem erfindungsgemäßen Hilfesystem ist die Dialoginitiative gemischt. Der Benutzer kann sein Wissen über das System zur Beschleunigung der Bedienung nutzen und ermüdet nicht so schnell. Es ergeben sich daher Vorteile sowohl für den Endbenutzer des Systems als auch für den  
10 Systementwickler.

- Die Vorteile für den Systementwickler liegen insbesondere darin, dass das Hilfesystem während der Systemerstellung automatisch aus den Spezifikationen erzeugt wird und nicht  
15 gesondert programmiert zu werden braucht. Der Systementwickler muss lediglich Hilfe-prompts in vorgefertigte Hilfeslots einfügen. Dazu ist nur minimaler Aufwand erforderlich.

- 20 Durch die Definition von Grammatiken für Eingaben werden verschiedene sprachliche Eingaben für einen Befehl möglich. Das System wird flexibel hinsichtlich unterschiedlicher Ausdrucksweisen unterschiedlicher Benutzer.

- 25 Zum Navigieren, wenn der Benutzer zurück will oder die Orientierung völlig verloren hat, kann noch ein Befehl "Zurück!" vorgesehen werden, der das System dazu veranlasst, von Zustand c) in Zustand b) zurück oder von Zustand b) in Zustand a) zu wechseln.

30

Die beschriebene kontextsensitive Hilfe kann ferner hierarchisch strukturiert werden. Dies ist insbesondere hilfreich bei Systemen mit sehr vielen möglichen Transaktionen.

35

Eine derartige hierarchische Strukturierung erfolgt durch die Einführung von Subzuständen. Ein Subzustand hat einen Namen

und besteht aus einer Menge von Transaktionen und - optional  
- aus einer Menge von weiteren Subzuständen. Zu jedem  
Subzustand definiert man wiederum einen Prompt und eine  
Grammatik.

5

Man betrachte die folgenden Beispiele: Der Subzustand  
Auskunft besteht aus den Transaktionen Zugauskunft und  
Flugauskunft. Der Prompt für den Subzustand Auskunft lautet:  
"eine Auskunft erhalten". Die Grammatik für den Subzustand  
10 Auskunft, d.h. die möglichen sprachlichen Formen für die  
Eingabe, lautet z.B.: "Auskunft", oder "Auskunft erhalten".

Ähnlich sieht es für den Subzustand finanzielle Transaktion  
aus. Er besteht aus dem Subzustand Aktienhandel und den  
15 Transaktionen Geldüberweisung und Geldabhebung. Der Prompt  
für den Subzustand finanzielle Transaktion lautet:  
"Homebanking durchführen". Die Grammatik für den Subzustand  
finanzielle Transaktion lautet z.B.: "Homebanking".

20 Der Subzustand Aktienhandel besteht aus den Transaktionen  
Aktienkauf und Aktienverkauf. Der Prompt für den Subzustand  
Aktienhandel lautet: "mit Aktien handeln". Die Grammatik für  
den Subzustand Aktienhandel lautet z.B.: "Aktie".

25 Die Subzustände sollten dabei so definiert sein, dass jede  
Transaktion in höchstens einem Subzustand vorkommt. Ferner  
sollen die Grammatiken nicht überlappen d.h. eine Grammatik  
soll nur auf einen Subzustand hinweisen, da es sonst keine  
eindeutige Zuordnung zwischen Benutzeräußerung und Subzustand  
30 gibt.

Eine Möglichkeit, Subzustände zu definieren, wird im  
Folgenden anhand der Figuren näher erläutert. Im einzelnen  
zeigt:

35

Fig. 1 eine Menge von 21 Aktionen;

Fig. 2 eine Zerlegung der Menge gemäß Fig. 1 in 6 Subzustände der Ebene 1;

5 Fig. 3 eine Zerlegung der Menge gemäß Fig. 1 in Subzustände der Ebene 1 und Subzustände der Ebene 2; und

Fig. 4 eine Zerlegung der Menge gemäß Fig. 1 in Subzustände der Ebenen 1, 2 und 3.

10

Stufe 0:

Man betrachte zunächst eine Anwendung A mit den Transaktionen  $T_1, \dots, T_m$  (Fig. 1).

15

Im Folgenden wird diese Anwendung sukzessive hierarchisch zergliedert, wobei ausgehend von der untersten Stufe 0 schrittweise höhere Gliederungsebenen gebildet werden.

20 Stufe 1: Erzeugung von Subzuständen der Ebene 1:

Ist  $A_{11}$  eine Teilmenge von A mit mindestens 2 Elementen, so wird durch  $S_{11} := A_{11}$  ein Subzustand der Ebene 1 definiert. Ist  $A_{12}$  eine Teilmenge von  $A \setminus A_{11}$  mit mindestens 2 Elementen, wobei " $\setminus$ " für die Exklusionsoperation "ohne" steht, so wird durch  $S_{12} := A_{12}$  ein weiterer Subzustand der Ebene 1 definiert.

25

Ist  $A_k$  eine Teilmenge von  $A \setminus A_{11} \setminus A_{12} \setminus \dots \setminus A_{k-1}$  mit mindestens 2 Elementen, so wird durch  $S_k := A_k$  ein weiterer Subzustand der Ebene 1 definiert, wobei k eine natürliche Zahl ist.

30

Die gesamte Anwendung zerfällt somit in eine Anzahl von Subzuständen der Ebene 1 und in Transaktionen, die in keinem Subzustand vorkommen. (s. Fig. 2), denn es gilt nach Konstruktion  $A = S_{11} \cup S_{12} \cup \dots \cup S_k \cup A \setminus A_{11} \setminus A_{12} \setminus \dots \setminus A_{k-1} \setminus A_k$ , wobei "U" für die Vereinigung zweier Mengen steht.

35

Wir definieren nun die Aktionenrestmenge der Stufe 1  $AR_1 := A \setminus A_{11} \setminus A_{12} \setminus \dots \setminus A_{1k-1} \setminus A_{1k}$  und die Subzustandsmenge der Stufe 1  $S_1 := \{S_{11}, S_{12}, \dots, S_{1k}\}$ .

- 5 Durch die Einführung von Subzuständen hat man nun eine Sicht auf die Menge  $A$  gewonnen, in der sie als aus weniger Bestandteilen bestehend betrachtet werden kann, da jeder Subzustand jetzt nur noch als ein Bestandteil aufgefasst wird.

10

Stufe 2: Erzeugung von Subzuständen der Ebene 2:

- Ist  $A_{21}$  eine Teilmenge von  $AR_1$  und  $sub_{21}$  eine Teilmenge der in Stufe 1 gebildeten Subzustandsmenge  $S_1$  so wird durch  $S_{21} := sub_{21} \cup A_{21}$  ein Subzustand der Ebene 2 definiert, wenn diese Vereinigung mindestens 2 Elemente enthält.

- Ist  $A_{22}$  eine Teilmenge von  $AR_1 \setminus A_{21}$  und  $sub_{22}$  eine Teilmenge von  $S_1 \setminus sub_{21}$  so wird durch  $S_{22} := sub_{22} \cup A_{22}$  ein Subzustand der Ebene 2 definiert, wenn diese Vereinigung mindestens 2 Elemente enthält.

- Ist  $A_{2l}$  eine Teilmenge von  $AR_1 \setminus A_{21} \setminus A_{22} \setminus \dots \setminus A_{2l-1}$  und  $sub_{2l}$  eine Teilmenge von  $S_1 \setminus sub_{21} \setminus sub_{22} \setminus \dots \setminus sub_{2l-1}$ , so wird durch  $S_{2l} := sub_{2l} \cup A_{2l}$  ein Subzustand der Ebene 2 definiert, wobei  $l$  eine natürliche Zahl ist, wenn diese Vereinigung mindestens 2 Elemente enthält.

- Wir definieren nun  $AR_2 := AR_1 \setminus A_{21} \setminus A_{22} \setminus \dots \setminus A_{2l-1} \setminus A_{2l}$  als die Aktionenrestmenge der Stufe 2,  $SubR_1 := S_1 \setminus sub_{21} \setminus sub_{22} \setminus \dots \setminus sub_{2l-1} \setminus sub_{2l}$  als die Subzustandsrestmenge der Stufe 1 und  $S_2 := \{S_{21}, S_{22}, \dots, S_{2l}\}$  die Subzustandsmenge der Stufe 2.

- 35 Die gesamte Anwendung zerfällt nun in eine gewisse Anzahl von Subzuständen der Ebene 2, eine gewisse Anzahl von Subzuständen der Ebene 1 und eine Restmenge von Aktionen, die

keinem Subzustand angehören (s. Fig. 3).

Gegenüber der in Stufe 1 konstruierten Zerlegung von A hat die in Stufe 2 gebildete Zerlegung den Vorteil, dass sie  
5 gröber ist in dem Sinne, dass sie aus noch weniger Elementen besteht als die in Stufe 1 gebildete, da ein Subzustand der Ebene 2 jetzt nur noch als ein Bestandteil aufgefasst wird.

Stufe m: Erzeugung von Subzuständen der Ebene m:

10

Das Verfahren wird iterativ fortgesetzt. Hat man schon Subzustände der Ebenen bis m-1 erzeugt und dabei Restmengen an unverbrauchten Aktionen und an unverbrauchten Subzuständen der Ebenen 1 bis m-2 erhalten, so kann man durch  
15 Zusammenfassen von Subzuständen der Ebenen m-1 und unverbrauchten Subzuständen der Ebenen m-2, m-3, ..., 1 und restlichen Aktionen neue Subzustände der Ebene m bilden, wobei man bei jeder solchen Bildung die Restmengen an Aktionen und Subzuständen der Ebenen 1, 2, ..., m-1  
20 entsprechend verkleinert. Da sich bei jeder Zusammenfassung die Anzahl der Elemente verringern soll, müssen immer mindestens 2 Elemente zusammengefasst werden (s. Fig. 4 für m=3).

25 Das Verfahren kann nach Belieben nach jeder Ebene beendet werden und macht spätestens dann keinen Sinn mehr, wenn die Einteilung so grob ist, dass nur mehr 2 Elemente übrig geblieben sind.

30 Durch das Auferlegen einer hierarchischen Struktur wird der bisherige Zustand a) durch den neuen Zustand d) ersetzt, der da lautet: "Das System befindet sich in einem Subzustand".

In Zustand d):

35

Das System befindet sich in einem Subzustand der Ebene n. Der Startzustand des Systems wird als Subzustand der Ebene m

definiert, wobei  $m$  die größte natürliche Zahl ist, zu der noch ein Subzustand gebildet wurde. In Fig. 4 z.B. ist  $m = 3$ .

Nun müssen zwei Fragen beantwortet werden:

5

Frage 1: Wie verhält sich das System bei Hilfeanforderung, wenn Situation d) vorliegt?

Antwort:

10 Das System befinde sich in einem Subzustand der Ebene  $n$ .  
Dieser möge aus den Subzuständen  $S_{1n-1}, \dots, S_{k(n-1)n-1}$  der Ebene  $n-1$ , den Subzuständen  $S_{1n-2}, \dots, S_{k(n-2)n-2}$  der Ebene  $n-2$ , ..., den Subzuständen  $S_{11}, \dots, S_{k(1)1}$  der Ebene 1 und einzelnen Transaktionen  $T_1, \dots, T_{k(0)}$  bestehen. Dabei ist  
15  $k(.)$  eine Funktion, die zu jeder Zahl, welche die Ebene bedeutet, die Anzahl der zu dieser Ebenenzahl gehörenden Subzustände bestimmt.

Das System äußert dann:

20 "(Pauschaler Hilfeprompt):  
(Prompt von  $S_{1n-1}$ ),  
...,  
(Prompt von  $S_{k(n-1)n-1}$ ),  
(Prompt von  $S_{1n-2}$ ),  
25 ...,  
(Prompt von  $S_{k(n-2)n-2}$ ),  
(Prompt von  $S_{11}$ ),  
...,  
(Prompt von  $S_{k(1)1}$ ),  
30 (Prompt von  $T_1$ ),  
...,  
(Prompt von  $T_{k(0)}$ )"

Frage 2: Wie gerät das System in einen Subzustand und welches  
35 Systemverhalten kennzeichnet den Subzustand?

*Ersteres geschieht, indem der Benutzer in der Ebene, in der der Subzustand definiert ist, eine Äußerung tätigt, welche durch die Grammatik analysiert wird und als Auswahl des Subzustands verstanden wird.*

5

Ist  $m$  die größte natürliche Zahl, zu der noch ein Subzustand gebildet wurde (Z.B.  $m = 3$  in Fig. 4), so besteht der Anfangszustand des Systems aus Subzuständen bis maximal Ebene  $m$  und restlichen Aktionen. Wird nun die Äußerung des Benutzers durch die Grammatik eines Subzustand der Ebene  $n < m$  analysiert, so springt das System in diesen über.

Das Systemverhalten des dann erreichten Subzustand bzw. seiner Elemente ist dadurch gekennzeichnet, dass noch keine Transaktion ausgewählt wurde, jetzt aber nur noch die im Subzustand direkt oder in einem darunter liegenden Subzustand enthaltenen Transaktionen zur Verfügung stehen.

Wird durch eine Benutzeräußerung ein Subzustand erreicht, so äußert das System je nach Wert der oben erwähnten Dialogschwelle  $D$  die zur Verfügung stehenden Subzustände und Transaktionen nach dem oben beschriebenen Muster, d.h. entweder in der Form "(Pauschaler Hilfeprompt)" gefolgt von der Aufzählung der Transaktionsprompts oder durch "(Frageprompt) + (Transaktionsprompt von  $T_i$ )".

Das Verfahren des Auferlegens von Subzuständen ähnelt der Vorgehensweise der IVR-Systeme, jedoch ist im Unterschied zu diesen ein direkter Zugriff über mehrere Zwischenebenen hinweg möglich. Alle ursprünglichen Möglichkeiten des Systems bis zur direkten Angabe einer Transaktion mit ihren Parametern bleiben erhalten. Die Strukturierung mittels Subzuständen stellt lediglich eine Hilfestellung für ungeübte Benutzer dar.

35

Die Vorteile für den Benutzer liegen insbesondere darin, dass die Hintergrundapplikation durch die in die ursprünglich

flache Struktur nachträglich eingebrachte hierarchische Strukturierung an Selbstbeschreibungsfähigkeit gewinnt. Bedienen ohne Bedienungsanleitung wird dadurch noch eher möglich.

5

Für ungeübte Benutzer ergibt sich der Vorteil, dass sich das System auf explizite Anforderung lokal wie ein IVR-System verhalten kann. Die Möglichkeiten des Benutzers auf direkten Zugriff bleiben jedoch erhalten. Damit entstehen Systeme, die von geübten und ungeübten Benutzern gleichermaßen gut bedient werden können.

Zum Navigieren kann wiederum der Befehl "Zurück" vorgesehen werden, der das System dazu veranlasst, von Zustand c) in Zustand b) zurück oder von Zustand b) in Zustand d), bei dem sich das System in einem Subzustand befindet, bzw. innerhalb des Zustands d) von einer Ebene in eine höhere Ebene zu wechseln.

Für die Dialogmaschine Diane wurde eine kontextsensitive Hilfe nach obiger Beschreibung implementiert. Wird als konkrete Anwendung die oben beschriebene Hintergrundapplikation benutzt, so könnten folgende Dialoge geführt werden (falls eine Dialogschwelle von  $D = 2$  definiert wird):

Dialog 1: (zeigt den schrittweisen Abstieg in der Hierarchie.  
Der ungeübte Benutzer navigiert nur mit dem Hilfefkommando bzw. äußert eine der vom System angebotenen Möglichkeiten)

30 System: Was kann ich für sie tun?  
Benutzer: Hilfe!  
System: Sie können eine Auskunft erhalten, Homebanking durchführen, einen Sender aufnehmen, eine  
35 Sendung aufnehmen, jemanden anrufen.  
Benutzer: Homebanking.



- System: Sie können mit Aktien handeln, Geld überweisen, Geld abheben.
- Benutzer: Mit Aktien handeln.
- System: Möchten Sie Aktien kaufen?
- 5 Benutzer: Nein.
- System: Möchten Sie Aktien verkaufen?
- Benutzer: Ja.
- System: Wie viel Stück?
- 10 Dialog 2: (zeigt den direkten Zugriff. Der geübte Benutzer kennt die Möglichkeiten des Systems und kann sich direkt äußern).
- System: Was kann ich für sie tun?
- Benutzer: Ich möchte Aktien kaufen.
- 15 System: Wie viel Stück?
- Dialog 3: (zeigt den Sprung auf einen Subzustand)
- System: Was kann ich für sie tun?
- Benutzer: Ich möchte Homebanking betreiben.
- 20 System: Sie können mit Aktien handeln, Geld überweisen, Geld abheben.
- Benutzer: Ich möchte 200 Siemens Aktien kaufen.
- System: Zu welchem Limit?
- 25 Dialog 4: (zeigt die Parameterhilfe, wenn die Parametergrammatik nicht die Generierungseigenschaft hat)
- System: Was kann ich für sie tun?
- Benutzer: Ich möchte 200 Siemens Aktien kaufen.
- 30 System: Zu welchem Limit?
- Benutzer: Hilfe!
- System: Sagen Sie einen Geldbetrag in Euro.
- Dialog 5: (zeigt die Parameterhilfe, wenn die Parametergrammatik die Generierungseigenschaft hat)
- 35 System: Was kann ich für sie tun?

Benutzer: Ich möchte Aktien kaufen.

System: Wie viel Stück?

Benutzer: 100.

System: Von welcher Gesellschaft?

5 Benutzer: Hilfe!

System: Sagen eine der folgenden Möglichkeiten:  
Adidas, BASF, Bayer, Commerzbank, Deutsche  
Bank, Dresdner Bank, Epcos, ... (listet alle  
Möglichkeiten auf).

10

## Patentansprüche

1. Verfahren zum Bereitstellen von Hilfe-Informationen für einen Benutzer in einem Sprachdialogsystem zur Bedienung  
5 einer Hintergrundapplikation,  
wobei die Hintergrundapplikation nach folgenden Prinzipien modelliert ist:  
P1) die Hintergrundapplikation kann als eine endliche Menge von Transaktionen T1, T2, ..., Tn aufgefasst werden;  
10 P2) jede Transaktion besitzt eine endliche Menge von zur Ausführung der Transaktion benötigten Parametern;  
P3) zu jedem Parameter gehört eine Grammatik, die dazu dient, den Parameter im Dialog zu erfassen;  
wobei das Sprachdialogsystem die folgenden Zustände einnehmen  
15 kann:  
Zustand a): es ist noch keine Transaktion ausgewählt und die Transaktionen T1, T2, ..., Ti sind noch möglich;  
Zustand b): es wurde eine Transaktion ausgewählt, es wurden aber noch nicht alle Parameterwert zu dieser Transaktion  
20 eingegeben;  
dadurch gekennzeichnet,  
dass für jede Transaktion ein Transaktionsprompt gespeichert wird;  
dass für jeden Parameter ein Hilfeprompt gespeichert wird;  
25 dass ein globales Hilfefkommando zum Aufrufen einer Hilfe gespeichert wird;  
dass das Sprachdialogsystem nach Erkennen des globalen Hilfefkommandos die im jeweiligen Zustand und Kontext gültigen Optionen durch die gespeicherten Prompts ausgibt,  
30 wobei im Zustand a) mindestens ein Transaktionsprompt ausgegeben wird; und  
wobei im Zustand b) mindestens ein Hilfeprompt ausgegeben wird.
- 35 2. Verfahren nach Anspruch 1,  
dadurch gekennzeichnet,  
dass der für jeden Parameter gespeicherte Hilfeprompt

entweder alle möglichen Werte dieses Parameters aufzählt oder die Form angibt, in der ein Wert für den Parameter einzugeben ist.

- 5    3. Verfahren nach Anspruch 1 oder 2,  
dadurch gekennzeichnet,  
dass im Zustand a)
- entweder alle möglichen Transaktionen unter Ausgabe der jeweiligen Transaktionsprompts ausgegeben werden,
  - 10 - oder der Benutzer mittels einer mit ja oder nein zu beantwortenden Frage unter Ausgabe des jeweiligen Transaktionsprompts für jede zur Verfügung stehende Transaktion einzeln gefragt wird, ob er diese Transaktion ausführen will, wobei der Benutzer, in dem Zustand des
  - 15 Systems, in dem das System auf eine Antwort auf die ja-nein-Frage wartet, als Antwort auf das globale Hilfefkommando einen Hinweis zur Beantwortung der ja-nein-Frage erhält.
- 20    4. Verfahren nach mindestens einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
dass ein pauschaler Hilfeprompt gespeichert wird; und  
dass der Benutzer durch Äußern des globalen Hilfefkommandos
- 25 die im Zustand a) gültigen Optionen vom Sprachdialogsystem durch eine Kombination des pauschalen Hilfeprompts und der Transaktionsprompts vermittelt bekommt.
5. Verfahren nach mindestens einem der vorhergehenden
- 30 Ansprüche,  
dadurch gekennzeichnet,  
dass ein Möglichkeitsprompt gespeichert wird, welcher dazu dient, alle im jeweiligen Kontext des Zustands b) möglichen Werte für den jeweiligen Parameter aufzuzählen.

6. Verfahren nach mindestens einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
dass für jede mögliche Eingabe eine Grammatik gespeichert  
5 wird.

7. Verfahren nach mindestens einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
10 dass die zur Verfügung stehenden Transaktionen hierarchisch geordnet werden.

8. Anordnung zum Bereitstellen von Hilfe-Informationen für einen Benutzer in einem Sprachdialogsystem zum Bedienen einer  
15 Hintergrundapplikation,  
wobei die Hintergrundapplikation nach folgenden Prinzipien modelliert ist:

P1) die Hintergrundapplikation kann als eine endliche Menge von Transaktionen T1, T2, ..., Tn aufgefasst werden;  
20 P2) jede Transaktion besitzt eine endliche Menge von zur Ausführung der Transaktion benötigten Parametern;  
P3) zu jedem Parameter gehört eine Grammatik, die dazu dient, den Parameter im Dialog zu erfassen;  
wobei das Sprachdialogsystem die folgenden Zustände einnehmen  
25 kann:

Zustand a): es ist noch keine Transaktion ausgewählt und die Transaktionen T1, T2, ..., Ti sind noch möglich;  
Zustand b): es wurde eine Transaktion ausgewählt, es wurden aber noch nicht alle Parameterwert zu dieser Transaktion  
30 eingegeben;

gekennzeichnet durch  
Mittel zum Speichern jeweils eines Transaktionsprompts in Zuordnung zu jeder Transaktion;  
Mittel zum Speichern jeweils eines Hilfeprompts in Zuordnung  
35 zu jedem Parameter;  
Mittel zum Erkennen eines globalen Hilfefkommandos;

Mittel zum Ausgeben der im jeweiligen Zustand und Kontext gültigen Optionen mittels der gespeicherten Prompts in Abhängigkeit vom Erkennen des globalen Hilfefokommandos, wobei die Mittel zum Ausgeben derart ausgebildet sind, dass

5 sie im Zustand a) mindestens ein Transaktionsprompt ausgegeben; und

wobei die Mittel zum Ausgeben derart ausgebildet sind, dass sie im Zustand b) mindestens einen Hilfefrompt ausgegeben.

## Zusammenfassung

### Bereitstellen von Hilfe-Informationen in einem Sprachdialogsystem

5

Ein Benutzer kann sich an jeder Stelle des Dialogs durch Aufrufen der kontextsensitiven Hilfe über die generelle Frage "Was ist möglich?" oder "Hilfe!" einen Überblick über die gerade gültigen Möglichkeiten verschaffen. Die

10

Hilfeinformationen werden durch eine Sprache erzeugt, die zur Modellierung der zugrundliegenden Hintergrundapplikation dient, indem die Sprache zum jeweiligen Kontext gehörige Hilfeprompts enthält. Das System generiert aus den Hilfeprompts und dem Kontextwissen die jeweils angemessene

15

Hilfe.

1 / 4

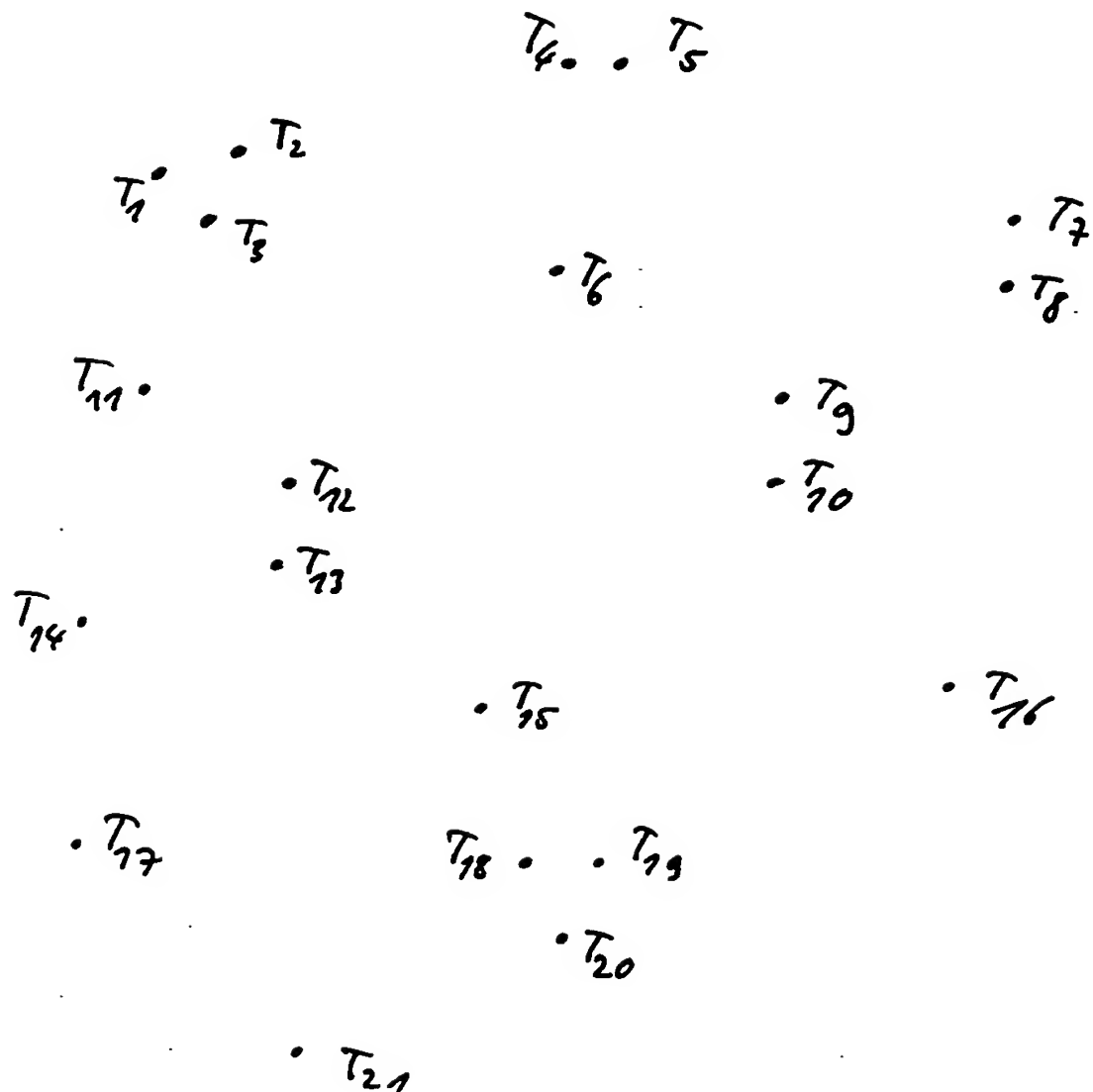
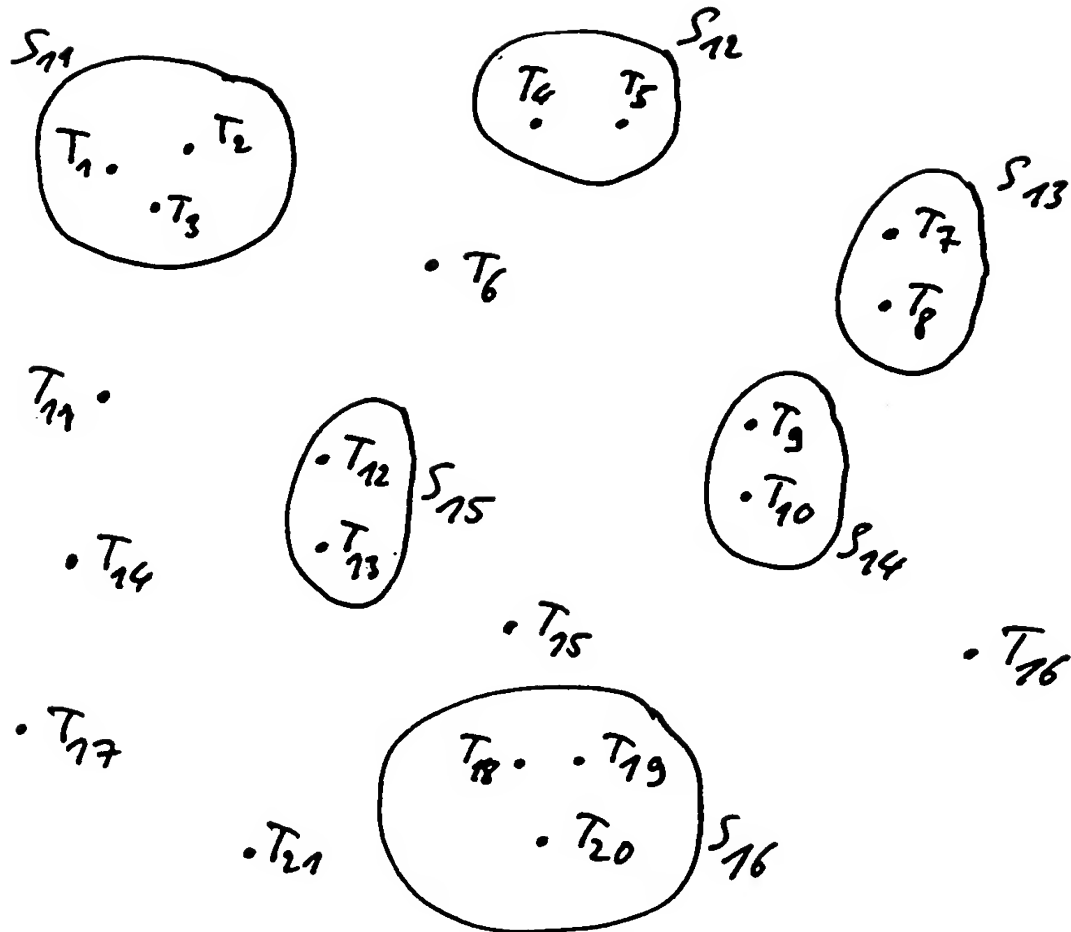


Fig. 1



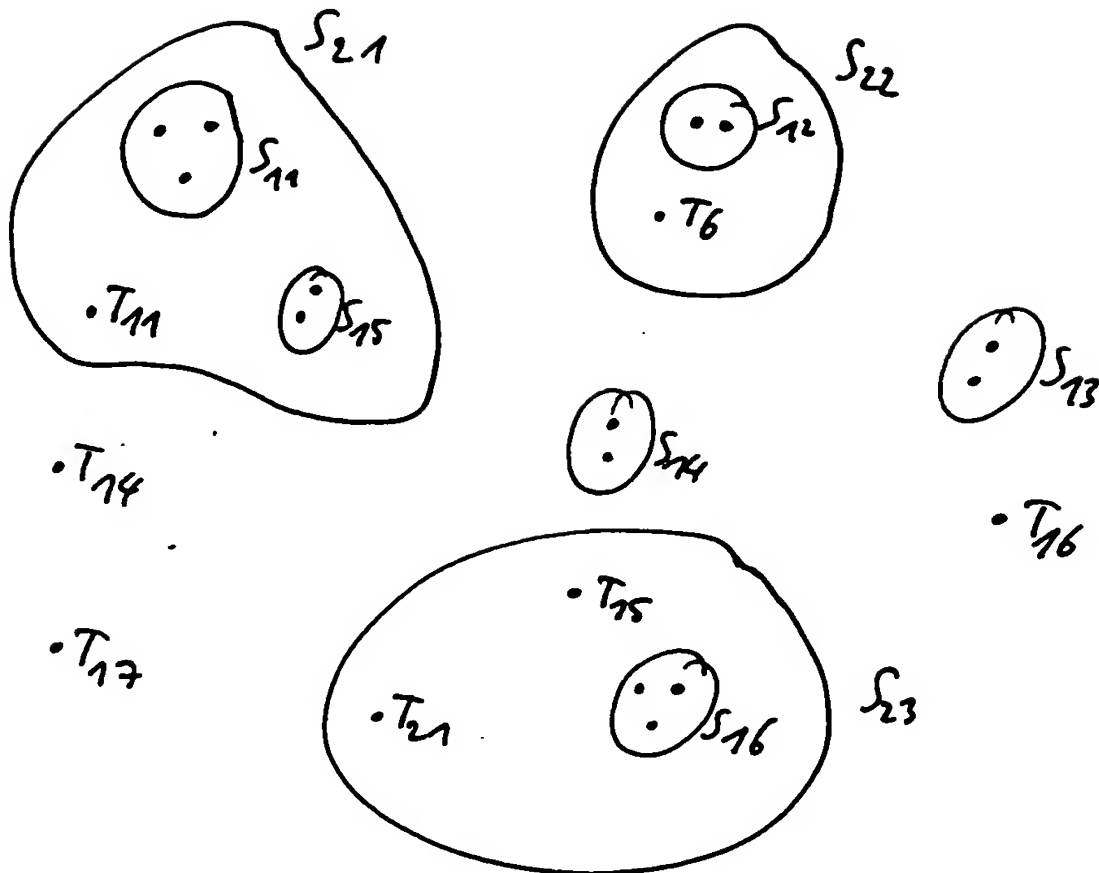
2 / 4



$S_{11} := A_{11} := \{T_1, T_2, T_3\}$        $S_{14} := A_{14} := \{T_9, T_{10}\}$   
 $S_{12} := A_{12} := \{T_4, T_5\}$        $S_{15} := A_{15} := \{T_{12}, T_{13}\}$   
 $S_{13} := A_{13} := \{T_7, T_8\}$        $S_{16} := A_{16} := \{T_{18}, T_{19}, T_{20}\}$   
 $AR1 = \{T_6, T_{11}, T_{14}, T_{15}, T_{16}, T_{17}, T_{21}\}$   
 $S1 = \{S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}\}$

Fig. 2

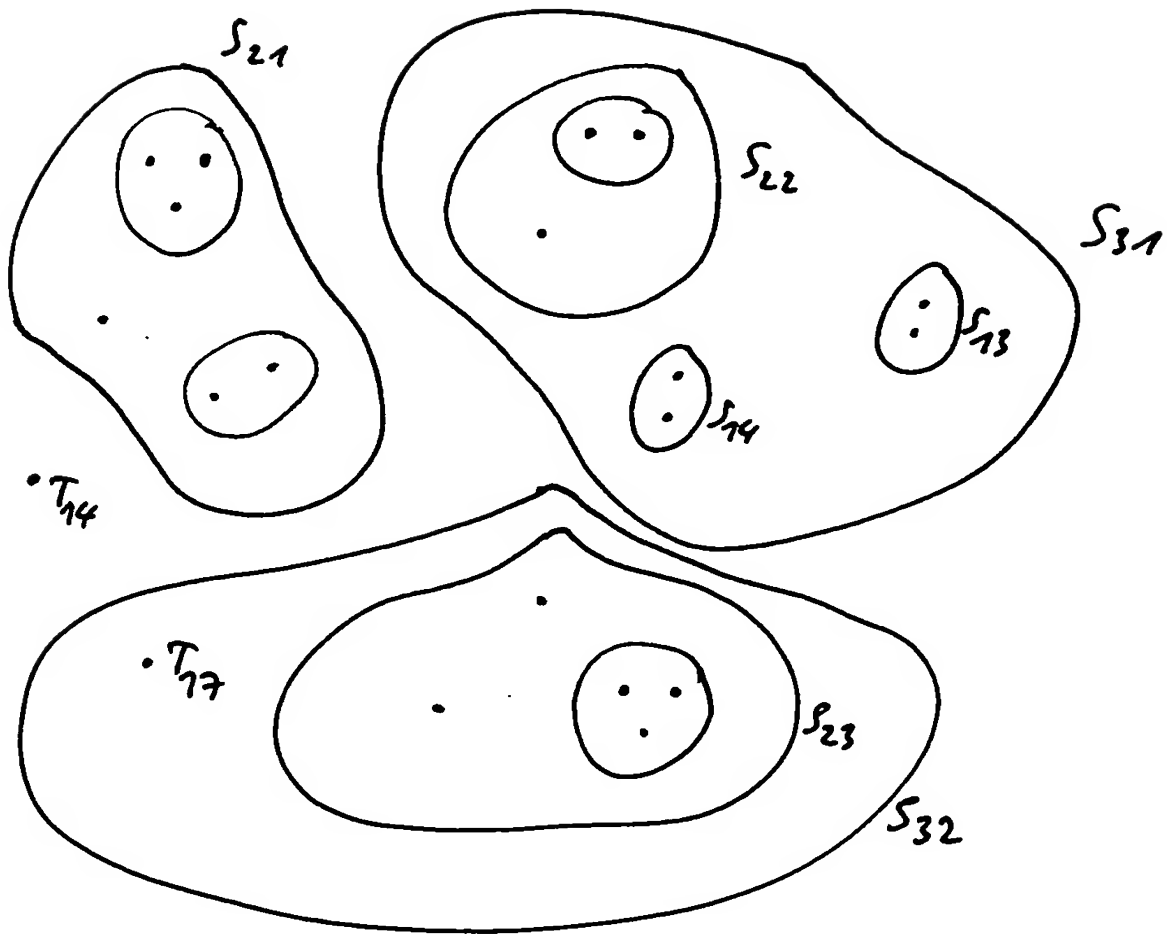
3 / 4



$A_{21} := \{T_{11}\}$ ,  $sub_{21} := \{S_{11}, S_{15}\}$ ,  $S_{21} = \{S_{11}, S_{15}\} \cup \{T_{11}\}$   
 $A_{22} := \{T_{16}\}$ ,  $sub_{22} := \{S_{12}\}$ ,  $S_{22} = \{S_{12}\} \cup \{T_{16}\}$   
 $A_{23} := \{T_{15}, T_{21}\}$ ,  $sub_{23} := \{S_{16}\}$ ,  $S_{23} = \{S_{16}\} \cup \{T_{15}, T_{21}\}$   
 $AR_2 = \{T_{14}, T_{16}, T_{17}\}$   
 $sub_{R1} = \{S_{13}, S_{14}\}$   
 $S_2 = \{S_{21}, S_{22}, S_{23}\}$

Fig. 3

4 / 4



$A31 := \{T16\}, \text{ sub31} := \{S13, S14, S22\},$   
 $S31 = \{S13, S14, S22\} \cup \{T16\}$   
 $A32 := \{T17\}, \text{ sub32} := \{S23\}, S32 := \{S23\} \cup \{T17\}$   
 $AR3 = \{T14\}, \text{ subR2} = \{S21\}, S3 = \{S31, S32\}$

Fig. 4